Programming 2
Tutorial and Practical 1

This week's tutorial and practical sessions concentrate on refreshing the knowledge you have gained from your earlier studies in Java (i.e. Programming 1 course).

Tutorial Questions

1. What does visibility mean, in the context of object properties? What do the visibility modifiers influence?

2. If there were 10 instances of a class that had a `static` property, how many copies of that property exist in memory? What if there were none?

3. The `final` modifier has slightly different meanings, depending on the property it is used with. Explain the meaning of final for primitives, reference types, methods and classes.

4. Answer true or false to the following questions:

    a. The value of a `final` primitive can be changed.
    b. The values inside a `final` object cannot be changed.
    c. A `final` method cannot be inherited.
    d. A `final` method can be inherited, but cannot be overridden.
    e. A `final` class cannot be extended.
    f. A `final` class can have non-`final` data.
    g. Methods of a `final` class are all implicitly `final`.

5. Explain the differences between overridden and overloaded methods. How can the Java interpreter differentiate between overloaded methods? When can't a method be overridden?

Practical Session

Weblearn Test

There is a Weblearn test (worth 2% of the course mark) related to Programming 1 revision to complete. This test should take no longer than 45-50 minutes to complete.

Once you have completed the test do the practical questions below.

Practical Questions

1. Write a driver class for the `CatalogueItem` class discussed in the online lecture notes for this week. The program should prompt for information, create a new `CatalogueItem` object from that data, and then print all the item information to the screen (code, description, price and tax component) using the appropriate accessor methods. For example:

```
Enter a catalogue code: TV140
Enter a description: Television
Enter a price: 410.00

The item details are:
Code:   TV140
Desc.:  Television
Price:  451.00
Tax:    41.00
```

2. From your code in the earlier question, add support for the `ImportedItem` class discussed in this week's lecture material.

   The driver class should this time prompt for a country of origin as well; if this prompt is left blank, the driver class should create a `CatalogueItem` object. If a country is entered, an `ImportedItem` object should be created. (There should only be one `CatalogueItem` object declared; if an `ImportedItem` is entered, use polymorphism to store it in the `CatalogueItem` reference.)

   Once the data is stored, print out what type of object resides in the reference, and print out all relevant details (i.e. don't show country of origin at all if a `CatalogueItem` is being displayed. You can use the `instanceof` operator and type-cast to get to any unique properties of `ImportedItem`.)

   Note: later on in this course, you will learn how to structure object and functionality hierarchies in a more elegant fashion, such that the use of `instanceof` and type-casts can be avoided.

3. Modify the program further to read in three sets of data, and use an array of `CatalogueItem` objects to store them. After data entry has completed, it should display all the items in a table-like view. The program should be written in such a way that the array size can be changed at the array declaration line of the code, with no additional code modification required.